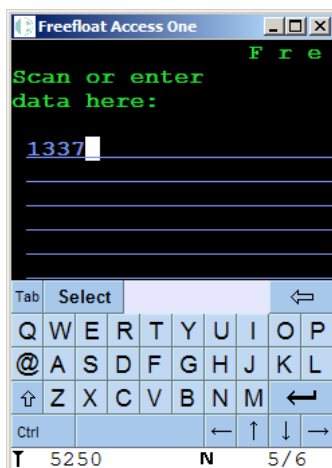
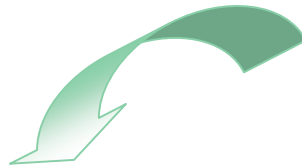
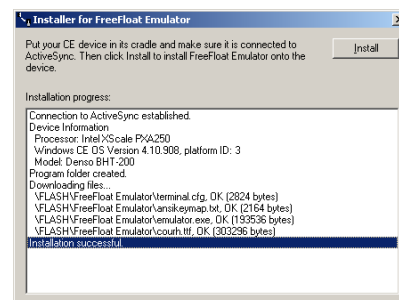
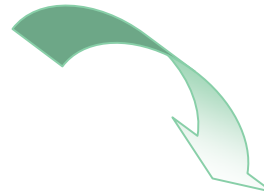
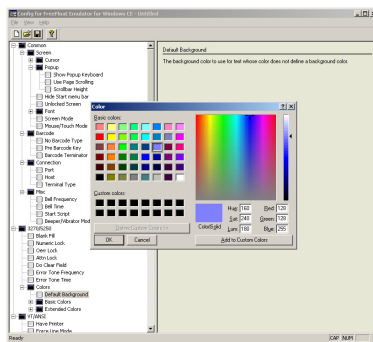




# freefloat Access\*One

## version 1.4.35



GO!

Contacting Freefloat.....	4
Introduction.....	5
System Requirements.....	5
Supported Windows Devices.....	5
Terminal Emulation.....	6
Product Features.....	6
Emulator Features.....	7
5250 Features.....	8
Getting Started.....	9
A Typical Installation.....	9
Step 1, testing the host connection from your PC.....	9
Step 2, configure and install on CE device.....	9
Step 3, licensing the emulator.....	10
The Emulator.....	13
Screen Layout.....	13
Emulator Text Area.....	14
Keyboard.....	15
Status Area.....	16
The Splash Screen.....	17
Splash Screen Mode.....	17
Locating the cursor, centering the screen around it.....	18
Scrolling the screen without moving the cursor position (moving the viewport).....	18
Moving the cursor (insertion point).....	18
Scanner.....	19
Scanner On/Off Control in 3270/5250 emulation.....	19
Scanner On/Off Control in ANSI emulation.....	19
Display Character Translation.....	20
Screen Modes.....	22
Standard Mode.....	22
Locked Mode.....	22
Center Cursor Mode.....	22
Screen Cursor Mode.....	23
Corner Mode.....	23
Full screen Mode.....	23
VT/ANSI.....	24
Line Mode.....	24
Attribute Mapping.....	25
Printing.....	26
Barcode Suffix.....	27
Barcode Prefix.....	27
Keystrokes in a Barcode.....	28
Editing Barcode Data.....	29
Simple Replacement.....	29
Advanced Barcode Editing.....	30
Logging.....	30

The Configuration Application .....	31
The Screen Layout .....	31
Option Types.....	32
The CEInstaller Application .....	33
What is transferred during installation and where is it? .....	34
Keyboard Mapping .....	35
How to map a key .....	35
How to start a script from a key mapping.....	36
Key Table.....	36
Configurable Keyboard.....	40
Scripting.....	41
Hotspot Scripts.....	41
Starting and Stopping Scripts.....	42
Script Commands.....	43
Appendix A – Data Transmitted by Various Keys in VT/ANSI Mode .....	48
Appendix B – Model Specifics .....	49
Denso BHT-200 .....	49
Denso BHT-400 .....	52
Symbol PPT8846 .....	52
Machines that have a sound device capable of playing wav-files .....	52
PSC Falcon 44xx with monochrome screen .....	52
Appendix C – Logging Network Data .....	54
Appendix D – EBCDIC Character Tables .....	55
Code Page 278, Swedish/Finnish.....	55
Code Page 37, USA/Canada .....	56
Appendix E – TN5250 Auto-signon .....	57
Appendix F – TN5250 Enhancement Settings.....	57
Appendix G – Command Line Parameters .....	58
Appendix H – WLinQ Bell integration.....	59
Configuration .....	59
Trigger Text .....	59
Text Row/Text Column .....	59
Cursor Row/Cursor Column .....	59
Data File.....	59
Sample Configuration .....	60
Appendix I – Version History .....	61

Copyright © 2003–2008 Freefloat AB

## **Contacting Freefloat**

### ***Office***

Freefloat AB  
P.O. Box 13101  
S-402 52 GÖTEBORG  
SWEDEN

### ***Support and Sales Information***

Forum      <http://forum.freefloat.com/>  
Web        <http://www.freefloat.com/>  
E-mail     <mailto:info@freefloat.com>

## Introduction

Freefloat Access\*One was developed to enable the many users of IBM mainframe(s) and various \*NIX and other ANSI/VT-compatible systems to seamlessly integrate these devices into their daily operations at a reasonable cost.

The emulator supports VT52, VT100 and VT220, TN3270, and TN5250.

Functions that can't be supported on certain devices due to hardware limitations are either ignored or mapped as appropriate.

## System Requirements

**Note!** The below requirements are for the PC, onto which the product is installed. The actual terminal emulator runs on the devices listed below.

- Windows XP/2000
- Minimum 16 MB memory, (the Windows emulator takes max 3 MB of memory)
- The disk space required is displayed by the installation program
- Microsoft ActiveSync v3.5 or later (for installing on a CE device, version 3.5, 3.7, 3.7.1, and 3.8.0 has been tested)
- Adobe Acrobat Reader 4.05 or later

## Supported Windows Devices

- Denso BHT-200 (BHT-2xxBW-CE and BHT-2xxQW-CE) and BHT-400 (BHT-4xxBW-CE)
- DLoG DNet X7 and DLoG XMDA
- Generic Pocket PC 2002 or later
- Generic Windows XP/2000
- Intermec CN2 (10 key) and CK31
- Opticon PHL7000 and PHL8000
- PSC Falcon 44xx and Falcon 4220
- Symbol PPT8846 (CE, 15 key), PPT8846 (Pocket PC, 15 key), MC3070, MC3090, MC50, MC70, and MC75
- Unitech HT660, PA950, PA960
- Åkerströms Trux (distributed by Åkerströms: [www.akerstroms.se](http://www.akerstroms.se))
- Trafficom Wave (distributed by Consafe Logistics: [www.trafficom.com](http://www.trafficom.com))

### ***Terminal Emulation***

- IBM 3278-2 (24 x 80 monochrome)
- IBM 3279-2 (24 x 80 color)
- IBM 5251-11 (24 x 80 monochrome)
- IBM 3179-2 (24 x 80 color)
- DEC VT52, VT100, VT220

### ***Product Features***

- Configuration application
- Installation program for installing the emulator and configuration on the CE device
- Freefloat Key\*One integrated, configurable layouts via Key\*One Designer
- Full screen emulator for demonstration and testing purposes enabling you to test the configuration, scripts, etc. before deploying onto the CE device
- The emulator for regular Windows can also be licensed and used as a product
- Host simulators for VT, 3270, and 5250 which enables you to test the emulator on the CE device without having access to a real host

## ***Emulator Features***

- Scripting language
- The ability to block the user from accessing the Windows CE shell
- Support for multi-session to the same or a different server with the same or different emulation type
- Variable font size, bold or normal
- Large user-configurable keyboard
- Solid or blinking cursor
- Hotspot support
- Keystrokes can be encoded in barcodes
- Color support in all emulations
- Fully customizable colors
- VT/ANSI attributes can be mapped to colors
- VT/ANSI big text (double width, double width/double height)
- Supported text attributes: Underline, Reverse, Intensify (Bold), Blink, Colors, Invisible, Column Separator (5250)
- Centering of the viewport on cursor and/or movement keys
- Manual or automatic centering of the viewport
- Viewport locking
- Fully customizable character translation between the host and the device
- Customizable key strings
- Real full-screen usage on CE device, even hides (optionally) the taskbar.
- Tight device integration (for example barcode reader, radio control, beeper & vibrator)
- Customizable mouse (touch screen) behavior
- A different kind of signal (beep and/or vibrator) controllable from the host
- VT/ANSI pass-through printing

## **5250 Features**

- Handles more than 256 fields on one screen correctly.
- Supports the 5250 Text Window definition with full attribute support, header, footer including attributes and orientation for header and footer.
- 100 % color and attribute support in both color and monochrome mode.
- Handles multi-line fields correctly, even fields that cover the entire screen.
- Full and correct field type/field properties support:

Alphabetic Only (also accepts international letters according to IBM's specifications)

Alphabetic Shift (actually means all characters are allowed)

Monocase (also uppercases international letters)

Signed Numeric

Numeric Only

Bypass

Field Exit Required

Mandatory Entry

Mandatory Fill

Right Adjust

Zero Fill

Space Fill

Dup Enable

Numeric Shift (actually the same as Alphabetic Shift)

Modified (the host can set the modified status of a field directly in a screen definition)



## Getting Started

Since you are reading this you have managed to install the product on a PC. A program group called **Freefloat Access One** was created on your Start menu.

The program group contains icons for configuring the various emulators (CE device specific ones and the one for regular Windows), the CEInstaller program which installs the emulator on a CE device as well as the configuration, support files, character mapping files, and optionally scripts.

There is also an icon for starting the emulator on your PC.

### A Typical Installation

Let's say you have a 3270 host. It's IP address is 192.168.1.3. It serves TN3270 connections on port 23. You also have a Denso CE device that you want to connect to the host.

#### Step 1, testing the host connection from your PC


1. Select the icon **Config Freefloat Access One** on the Start menu.
2. The configuration application is displayed, the configuration for the regular Windows version of the emulator is automatically opened.
3. In the configuration application, go to the option *Common\Connection\Host*.
4. Enter the IP address for your host (192.168.1.3).
5. Go to the option *Common\Connection\Emulation Type*
6. Select the emulation (TN3270 Color 24x80).
7. Choose File/Save to save the modified configuration.
8. Exit the configuration application.
9. Start the Freefloat Access One by selecting the icon **Freefloat Access One** on the Start menu.
10. Make sure you can connect to your host.

This must be done with a new CE device prior to installing the emulator on it:

*Start your CE device and configure your CE device's RF network card so it can associate with an access point. For example: SSID (network name), IP address (fixed or dynamic), WEP encryption keys or other encryption settings if applicable. Please see the documentation for the CE device on how to do this.*

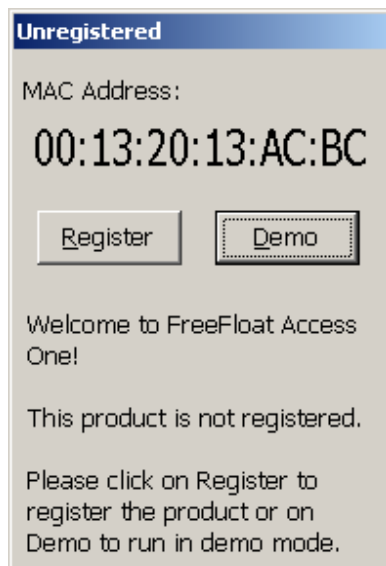
#### Step 2, configure and install on CE device

1. Select the icon **Config <manufacturer name>/<device name>** depending on what CE device you are using.
2. Go to the *Host* option and enter your host's IP address.
3. Go to the option *Emulation Type* and select the desired emulation.

4. Save the configuration and click on  in the toolbar of the configuration application. This will start the **CEInstaller** application. It installs the emulator on the CE device.
5. Start the CE device and make sure it is synchronized with your PC via Microsoft ActiveSync.
6. Click on the **Install** button.
7. After the emulator is installed, start it and check that the emulator connects to your host.

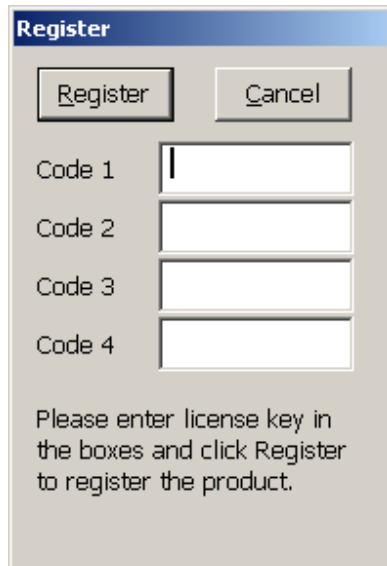
### Step 3, licensing the emulator

When the emulator is started on the CE device for the first time it is unlicensed. The following dialog box is displayed:



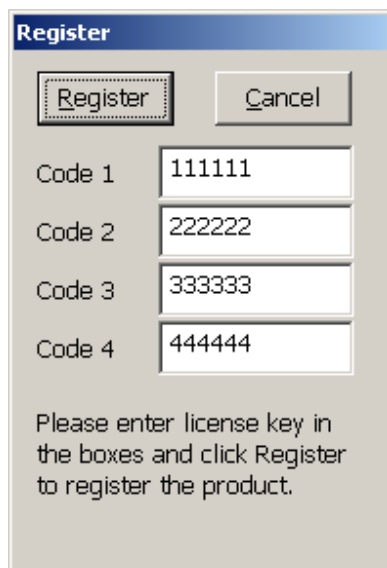
To license the emulator, write down the MAC address, contact your reseller or Freefloat to buy a license. You will then receive a license key. Until you have received the license key, you can run the emulator in demonstration mode. The demonstration mode has no limitations except that it times out and exits after 20 minutes.

When you receive the license key, click on the Register button in the dialog box above. The Register dialog box will be displayed:



The Register dialog box has a title bar labeled "Register". It contains two buttons at the top: "Register" and "Cancel". Below these are four text input fields labeled "Code 1", "Code 2", "Code 3", and "Code 4". The "Code 1" field contains a single vertical line cursor. At the bottom, there is a text instruction: "Please enter license key in the boxes and click Register to register the product."

Enter the digits from the license key into the boxes. Leave out the hyphens (-). If the license key you received was 111111-222222-333333-444444 your dialog should look like this:



The Register dialog box is shown with the license key digits entered. The "Code 1" field contains "111111", "Code 2" contains "222222", "Code 3" contains "333333", and "Code 4" contains "444444". The "Register" button is highlighted with a dashed border. The text instruction at the bottom remains the same: "Please enter license key in the boxes and click Register to register the product."

Click on the **Register** button to complete the registration process.

If you made a mistake when entering the license key, you will get the following message when you click the register button:



When you click the OK button you will be taken back to the Register dialog box.

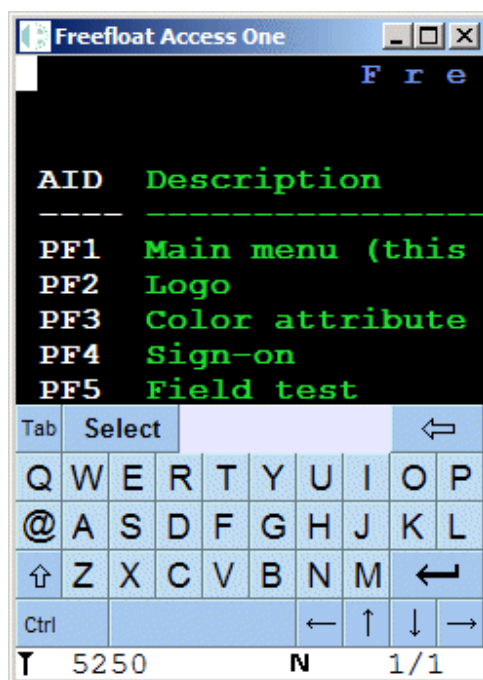
## The Emulator

This topic explains the different screen objects present in the emulator.

Please note that the window size of a CE device can be simulated in the regular Windows version of the emulator by setting the *Win32\Window Size* option to *Quarter VGA*.

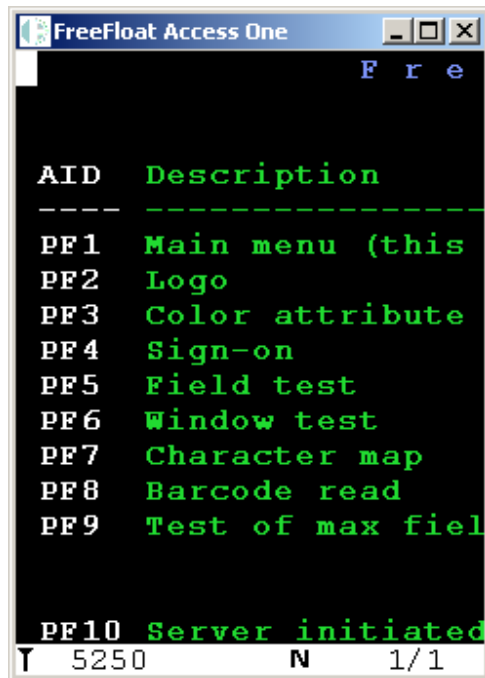
### Screen Layout

Here's how the emulator might look like on a Quarter VGA color device:



## Emulator Text Area

The emulator text area is the area above the status area.



Data from the host is in this area. Please note that the keyboard covers the text area when visible.

When you touch or click on the emulator text area the action depends on the option *Common\Screen\Mouse/Touch Mode*. See the configuration program for a detailed explanation of this parameter.

### Keyboard

The keyboard is displayed below the emulator text area:

Tab	Select							↩	
Q	W	E	R	T	Y	U	I	O	P
@	A	S	D	F	G	H	J	K	L
↑	Z	X	C	V	B	N	M	←	
Ctrl						←	↑	↓	→

There are canvases for characters (even international ones), function keys, movement keys, and some special control buttons.

If you click on the canvas background, i.e. not on a button, you can drag the canvas to a new position.

Use **Select** to select another canvas.

**Script** displays the *Select Script* dialog. Use it to start a script. If a script is running when you click on the **Script** button, the emulator will ask you if you want to stop the script.

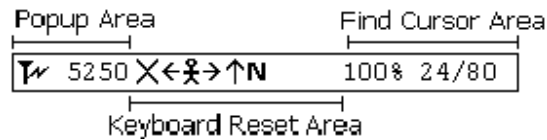
**Exit** will end the host session.

**Font +** **Font -** increases and decreases the font size used in the emulator text area.

**Note!** Changes to the font size is not saved to the configuration file. To make permanent changes to any option you need to use the configuration application and then transfer the new configuration file to the CE device.

**About** displays the version of the emulator:

## Status Area



**T** is the radio connection indicator.

*This indicator only works on Denso and Symbol (and some other models) handhelds, on other machines it displays as (T).*

If the flash disappears (T) the CE device has lost its connection with the Access Point.

**5250** this area displays the current emulation, ANSI, 3270 or 5250. In VT/ANSI emulation this text can be \_ANSI, LANSI or IANSI. If a space, the emulator is not in linemode, if L the emulator is in negotiated linemode, if l (lowercase L) it is forced local linemode.

Clicking on the *Popup Area* toggles the keyboard.

**X<F>** is the 3270/5250 operator error indicator.

<b>X&lt;F&gt;</b>	You tried to enter text outside a field
<b>X&lt;F&gt;NUM</b>	You tried to enter invalid characters into a numeric only field
<b>X&lt;F&gt;</b>	You tried to enter more data into a field than there is room for
<b>X&lt;F&gt;SYS</b>	The emulator is waiting for a host response
<b>X&lt;F&gt;</b>	Data has been received from the host and is being processed
<b>I</b>	Insert mode indicator
<b>A or N</b>	Input Mode indicator (only active on some models) Tells you whether the input mode is alpha or numeric. On a Denso BHT-200 you press the ALP key to toggle.

If the keyboard is locked, indicated by **X<F>**, **X<F>NUM**, and **X<F>**, you can click on the *Keyboard Reset Area* to unlock the keyboard.

**100%** is the battery level.

**24/80** is the cursor location indicator. To quickly locate the current cursor position (the active input), click on the *Find Cursor Area*.



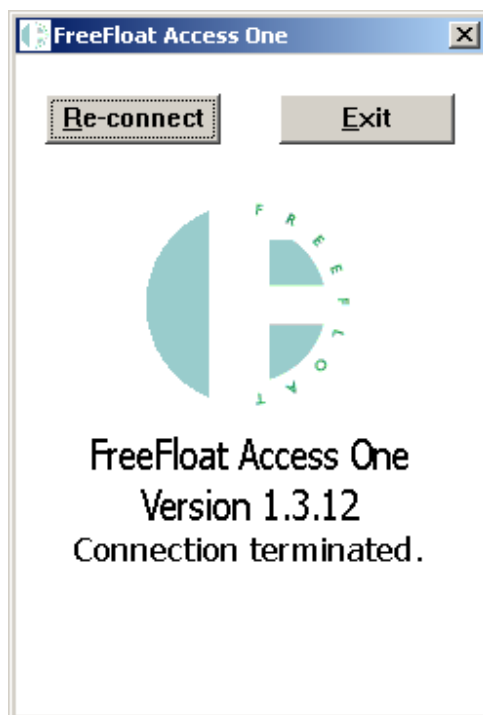
### The Splash Screen

When the connection to the host is terminated the Splash screen is displayed.

When the option *Forced Mode Password* has been given a value, a dialog box will appear. To be able to exit the emulator, the correct password must be entered.

The Splash screen is also used for displaying network error messages.

Here's how the Splash screen might look like:



### Splash Screen Mode

There are two different splash screen modes, the normal one with Re-connect and Exit buttons and a more specialized mode where the Reboot button has been made very large and centered at the bottom of the splash screen. In this specialized mode there is also a Reboot button which reboots Windows. The Reboot mode for the splash screen is only useable in Windows 2000/XP, not in Windows CE. Also it need VGA resolution or higher. The option that controls the splash screen mode is:

*Common\Screen\Splash Screen Mode*

Please note that the following three functions will NOT move the viewport when the screen has been locked with the script command LockWindow.

### ***Locating the cursor, centering the screen around it***

Due to the tiny screen of most CE devices, it is very easy to loose track of the cursor position. To quickly locate the current cursor position (the active input), the following shortcut method can be used.

Touch or click on the *Find Cursor Area*.

The emulator will try to center the viewport around the cursor, so the operator can see as much as possible of the screen at that position. This doesn't move the cursor, but moves the viewport to the cursor position.

### ***Scrolling the screen without moving the cursor position (moving the viewport)***

If the option *Mouse/Touch Mode* is set to Scroll, push and drag in the emulator text area.

If you need viewport scrolling from the Key\*One keyboard, define four keys that send Alt+arrow keys.

### ***Moving the cursor (insertion point)***

If the option *Center On Arrows* is enabled, the emulator will try to center the screen around the cursor, so the operator can see as much as possible of the screen at that position. This moves both the cursor and the viewport to the new cursor position.

If the option *Center On Arrows* is disabled, only the cursor moves, so if the cursor moves off the visible screen, the operator must use the function Locating the cursor, centering the screen around it described above, to locate the cursor.

On the keyboard canvas with arrow keys, use the arrow keys to move the cursor to the screen position desired.

If the CE device has hardware cursor keys, they can also be used to move the cursor.

### Scanner

*This section only applies to models where Freefloat Access\*One has direct control of the barcode device*

You can disable the option *Barcode Reader* if you are using a third-party barcode wedge software.

If the option *Barcode Reader* is disabled, the barcode parameters will have no effect.

All scanned data is internally converted to keystrokes, so any data scanned can also be input from the keyboard.

When data is entered into a field using the scanner the flow of events are as follows:

1. If the option *Barcode Prefix* defines a key string, that key string is processed first.
2. The scanned data is then entered into that field one character at a time.
3. If the field is too small to receive the scanned data, the emulator behaves in the same way as if too much data was entered from the keyboard.
4. If the option *Barcode Suffix* defines a key string, it is entered as if it came from the keyboard.

### Scanner On/Off Control in 3270/5250 emulation

By default, Freefloat Access\*One turns off the scanner when the cursor is outside any input field or when there are no input fields on the screen. But if for example you need to simulate AID keys (function keys) by reading a barcode, the scanner needs to be enabled at all times by enabling the parameter *Common\Barcode\Barcode Reader Always On*.

### Scanner On/Off Control in ANSI emulation

CSI % 0 S disables the scanner.

CSI % 1 S enables the scanner.

## Display Character Translation

### General Description

As delivered, the emulator is configured for the Swedish character set in 3270 or 5250 mode (Codepage 278), and in ANSI mode it uses the DEC Multinational character set (ISO/IEC 8859-1 Latin 1).

### ANSI Mode

ANSI characters from the wire are converted to Unicode using the ansi2uni.map table before being stored in the screen buffer for subsequent display. Characters entered via keyboard are translated using the uni2ansi.map table before they are put on the wire.

### 3270 or 5250 Mode

Input from the keyboard is converted to EBCDIC using the uni2ebc.map table before being put into the buffer. All characters in the buffer (whether they came from the host or from the keyboard) are converted from EBCDIC to Unicode using the ebc2uni.map table before being displayed on the screen.

You can change the default character mapping via one of the four mapping files:

Filename	Mapping function
uni2ebc.map	Unicode (from keyboard) to EBCDIC (in buffer) mapping.
ebc2uni.map	EBCDIC (in buffer) to Unicode (screen) mapping.
uni2ansi.map	Unicode to ANSI (VT220 Multinational) mapping.
ansi2uni.map	ANSI (VT220 Multinational) to Unicode mapping.

If any of these files exist, it will be used to change the default character mapping for the characters defined in that file. Only the characters defined in that file are changed from it's default mapping.

Sample contents of ansi2uni.map:

```
// ANSI2UNI translates from network ISO11 Swedish (7-bit ASCII)
// to screen Unicode
//
// ISO11 Swedish value arrived from the wire, Unicode value
// to display on screen

0x60,0xe9
0x7b,0xe4
0x7c,0xf6
0x7d,0xe5
0x7e,0xfc
0x5b,0xc4
0x5c,0xd6
0x5d,0xc5
0x40,0xc9
0x5e,0xdc
```

The corresponding file uni2ansi.map would normally reverse this mapping as shown below:

```
// UNI2ANSI translate from keyboard Unicode to network ISO11 Swedish
//
// Unicode value arrived from the keyboard, ISO11 Swedish value
// to send to host

0xe9,0x60
0xe4,0x7b
0xf6,0x7c
0xe5,0x7d
0xfc,0x7e
0xc4,0x5b
0xd6,0x5c
0xc5,0x5d
0xc9,0x40
0xdc,0x5e
```

## **Screen Modes**

The screen modes govern which portion of the larger terminal screen the emulator initially displays and how this viewport moves as the cursor moves.

### **Standard Mode**

This is the default mode and it begins with the window in the upper left corner of the larger terminal.

The user is free to move the viewport, either automatically as defined by the options *Center On Arrows*, *Center On Tab* and *Center On Backspace*, or by the two functions Locating the cursor, centering the screen around it

Or

Scrolling the screen without moving the cursor position (moving the viewport)

### **Locked Mode**

In locked mode the viewport is locked to the coordinates defined by the script command `SetOrigin`.

The default position is the upper left-hand corner of the display.

The viewport does not move around, and only the selected window area is visible.

Locked mode disables the windowing keys (arrow keys, tab, backtab and backspace) and only allows you to window around the physical display by the script command `SetOrigin`.

### **Center Cursor Mode**

Center cursor mode works best for applications that use the entire 80-character by 24-line logical display.

In this mode, the cursor remains in the center of the terminal's screen if possible.

As the cursor moves within the logical display, the emulator's viewport window moves to keep the cursor as centered as possible, showing as much as possible of the logical display on all sides of the cursor position.

## **Screen Cursor Mode**

Screen center cursor mode ensures that the cursor is always visible.

As the cursor moves within the logical display, the emulator's viewport window in increments of the viewport window's width and height. If, for example, the viewport can display 12 lines with 20 characters wide the screen will effectively be divided into 4x2 screen parts. The first part will contain column 1 to 20 of lines 1 to 12 and the last part will contain column 61 to 80 of lines 13 to 24.

When the cursor is moved (either by user or host) into one of these parts, the viewport will snap into the part.

To enable this mode set the parameter *Follow Cursor* to the value **Screen**.

## **Corner Mode**

Corner mode begins with the window in the upper left corner of the larger terminal. It tries to keep the cursor in the lower right corner of the display, showing as much as possible of the logical display above and to the left of the cursor position. Corner mode works best for applications that use the upper left corner of the logical screen.

As the cursor moves off the right or bottom edge of the radio terminal's display, the window moves to show the cursor. The distance moved and when movement should occur is determined by the options *Row Offset* and *Column Offset*.

## **Full screen Mode**

*Regular Windows version only.*

This mode use the entire screen, hides the taskbar.

Keys that bring up the Start menu or switch to other application are filtered, making it impossible to switch away from the emulator to any other program.

## **VT/ANSI**

### **Line Mode**

Do not activate this parameter for 3270 or 5250 emulation. 3270 and 5250 are both block-devices, where the whole screen is one big “line mode” and only transmits when Enter or a PF-key is pressed.

To activate ANSI Line mode, enable the option *Force Line Mode*.

In the default “character-at-a-time” mode, (also known as full duplex) every key press or character scanned from a barcode generates a network packet transmission to the host. The host then echoes the character back to the handheld, possibly pre-pending it with cursor positioning escape sequences and then the handheld displays the character on the screen. So, for every character pressed or scanned, the handheld generates one radio transmission and receives one or more packet(s).

For example, if we scan a 20-character barcode, the handheld will generate 20 packet transmissions, and receive at least 20 packets.

When the radio is active, the power consumption of the handheld is much higher, up to five times the normal consumption, so it is easy to see that if we can reduce the transmissions, the battery will last much longer.

Line mode will do just that.

As long as we remain on the current line, and the cursor position is to the right of the first input position, and no key is pressed that would move us away from the current line, (arrow keys, Page Up/Down, Enter or backspace beyond the first input position), every key pressed or character scanned is buffered in the handheld.

When a key is pressed that requires the collected data to be transmitted, all data is then sent as ONE packet.

Using the previous example, our 20 packet transmissions and receptions are reduced to one transmission, and usually also to one reception, a reduction by 20 times! Naturally, the battery now lasts much longer.

Not all applications work correctly in linemode, but the majority can benefit from this option.

Since all echoing of characters are done locally, password entries will be visible, because the handheld always echoes, and has no knowledge about the contents format.



However, this can be beneficial, since it is very easy to enter the wrong key using the sometimes awkward keyboards present on some handheld's, but the tiny screen usually prevents anyone else but the operator to read what was entered.

### Attribute Mapping

In VT/ANSI emulation, the attributes bold, reverse, underline, and blink can be mapped to a text color.

For example, to map the attribute bold to red, enable the parameter *VT/ANSI Emulation\Attribute Mapping\Map Bold*, and then make sure *VT/ANSI Emulation\Attribute Mapping\Bold Color* is set to red.

Mapped attributes have priorities in the order bold, reverse, underline, and blink. This means that if some text is both bold and underline and bold is mapped to red and underline to blue the text will be red because bold has higher priority than underline.

### Printing

Supported modes are:

#### Page print

Prints all information currently displayed on terminal screen.

	Octal	Decimal	Hex
ESC[0i	033 133 060 151	027 091 048 105	1B 5B 30 69

#### Controller print

Print is passed through the terminal and displayed only on the printer.

The terminal will pass all codes except the print on and off commands to the printer.

The terminal will not interact with the operating system during print.

Print on:	Octal	Decimal	Hex
ESC[5i	033 133 065 151	027 091 053 105	1B 5B 35 69

*Any data received after Print on is directed to the printer instead of to the screen.*

Print off:	Octal	Decimal	Hex
ESC[4i	033 133 064 151	027 091 052 105	1B 5B 34 69

### Configuration Settings

The setting *Port Name* can be any valid output device, the most common ones are LPTx: or COMx: .

If the port selected is a serial COM port, you must specify at least Baud, Parity, Data and Stop.

The settings *Port Settings* consist of the following (where each parameter is separated from the next by a blank/space):

Baud=*n* where *n* is any valid numeric baudrate.

Parity=*p* where *p* is **N** for None, **O** for Odd or **E** for Even parity.

Data= *n* where *n* is 5, 6, 7 or 8 databits.

Stop= *n* where *n* is 1 or 2.

Optional parameters are (default value if not specified shown as **bold**):

to= <b>on/off</b>	If on, printing will be aborted if no data could be printed for 60 seconds.
xon= <b>on/off</b>	Software handshake using the XON/XOFF protocol.
octs= <b>on/off</b>	Hardware handshake using the CTS line.
odsr= <b>on/off</b>	Hardware handshake using the DSR line.
dtr= <b>on/off</b>	Sets the DTR line (output) on/off.
rts= <b>on/off</b>	Sets the RTS line (output) on/off.
rts=tg	RTS is on only then transmitting.

*Models that support Barcode Suffix and Prefix include:*

- ☐ Denso BHT's
- ☐ PSC Falcon's
- ☐ Intermec handhelds
- ☐ Symbol handhelds
- ☐ Unitech PA950, PA960 and HT660
- ☐ Datalogic handhelds
- ☐ Psion Teklogix
- ☐ Opticon PHL7000
- ☐ SCC SC600
- ☐ PC's using Freefloat WLinQ for scanner integration

### **Barcode Suffix**

The option *Barcode Suffix* is the key string to append to the end of the scanned barcode. If it is defined, those keys will be acted on as if the corresponding keys had been pressed.

If no terminator is required, leave the option empty.

The two most frequently used suffixes for this option in 3270 or 5250 mode are <enter> (to send the currently scanned data to the host) and <tab> (to advance to the next input field).

### **Barcode Prefix**

The option *Barcode Prefix* is the key string to prepend to the beginning of the scanned barcode. If it is defined, those keys will be acted on as if the corresponding keys had been pressed.

If not required, leave the option empty.

The most frequently used key for this option in 3270 or 5250 mode is <home> (to move to the first input field). <clearfield> can be added if you would like the field cleared (only in 3270 and 5250).

## ***Keystrokes in a Barcode***

Models that support this function are:

- ☐ Denso BHT's
- ☐ PSC Falcon's
- ☐ Intermec handhelds
- ☐ Symbol handhelds
- ☐ Unitech PA950, PA960 and HT660
- ☐ Datalogic handhelds
- ☐ Psion Teklogix
- ☐ Opticon PHL7000
- ☐ SCC SC600
- ☐ PC's using Freefloat WLinQ for scanner integration

You can embed keystrokes in barcodes. To be able to do this, the barcode type must be able to encode the characters \, <, and >. This is possible in Code 128 and most 2D codes.

For example, you want to encode the text 123 followed by the TAB key and then ABC. Encode the following data in a barcode to accomplish this:



123\<tab>ABC

If you want to encode the character sequence \< you need to escape both characters with \ giving the barcode contents \\ \<.

## **Editing Barcode Data**

### **Simple Replacement**

A simple string replacement can be done on barcode data. You do this by specifying replacements in the file BCREPLACE.TXT. This file contains rules in a certain format that specify what replacements should be performed. The file should be placed in the folder for the device you are using, for example:

*C:\Program Files\Freefloat\Freefloat Access One\BHT400*

The format of a replacement rule is:

<substring 1>=<replacement string 1>, <substring 2>=<replacement string 2>, ...

The file BCREPLACE.TXT can have multiple rules, one on each line. If <substring 1> is found in the barcode then that line will be used for replacements. For example:

```
"/A" = "<f1>", "<enter>" = ""  
"/B" = "<f2>", "<enter>" = ""  
"/C" = "<f3>", "<enter>" = ""
```

If a barcode containing the text /B is read the rule on the second line will be matched. Then /B in the barcode will be replaced with the key mnemonic <f2>. Then the substring <enter> will be replaced with an empty string. <enter> comes from the barcode suffix as specified in the configuration. Then the processing stops.

In other words, reading a barcode containing /B will result in F2 being pressed without the barcode suffix (<enter>). If <enter> would not be filtered by the rules F2 and then Enter would be sent to the host.

**Note!** All strings must be enclosed in quotes. Whitespace outside quotes are ignored.

### Advanced Barcode Editing

Freefloat Access\*One can be configured to launch an external program that does advanced barcode formatting. To do this you need to write an application that reads a text file, formats it, and writes back the modified data. Please note that the format is Unicode (little endian) without the Unicode BOM (Byte Order Mark). In other words "ABC" would be encoded as:

```
41 00 42 00 43 00
```

You specify the program to be launched via the setting:

*Common\Barcode\External Processor\Executable*

If you, for example, specify *BarcodeEdit.exe*, Access\*One will try to launch a program with that name. It looks for the program in the installation folder. You may also specify an absolute path to the external program.

You also need to specify the temporary file for storing the barcode via the setting:

*Common\Barcode\External Processor\Data File Name*

If you specify a temporary filename without specifying a program to launch, that file will contain the last scanned barcode data, which sometimes can be useful for determining exactly what data a certain barcode contains.

### Logging

The emulator maintains a log file where information about certain event(s) and error conditions are registered.

The start time and emulator version are always registered each time the emulator starts.

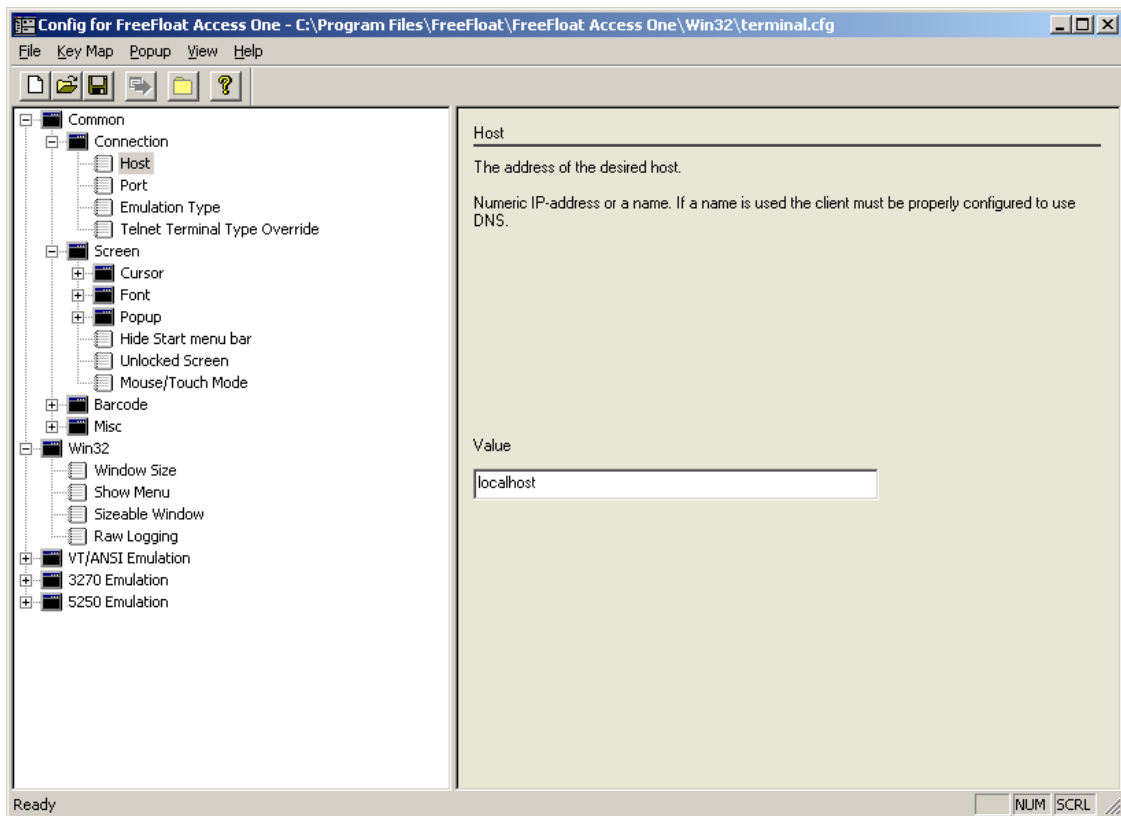
The log file is trimmed to the last 100 lines.

The name of log file is LOG.TXT and it's located in the root of the file system of the CE device. When running Freefloat Access One on regular Windows, the log file is placed in same directory as the executable.

## The Configuration Application

Use the Configuration Application to configure the emulator to your preferences. The emulator will only load it's settings from a configuration file called Terminal.cfg.

### The Screen Layout




In the left window pane you select the option you want to change or inspect. When you click on an option the option's value is displayed in the right window pane together with a description of the option.

## Option Types

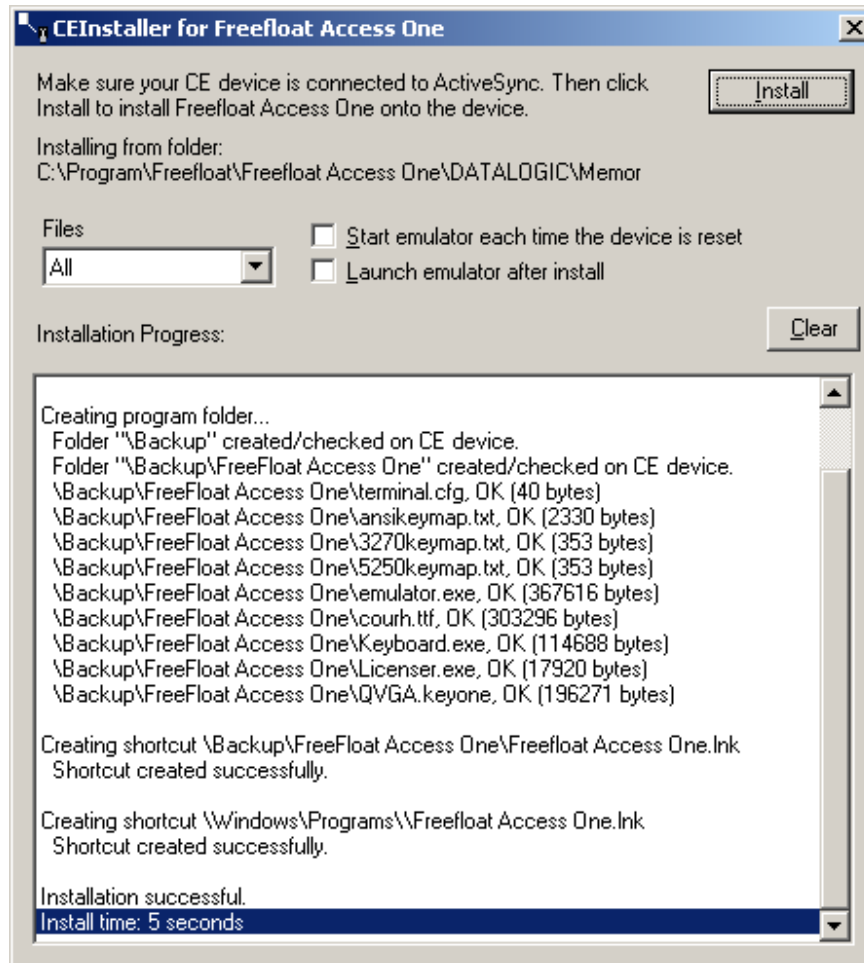
Type	Description	Sample
Toggle	An option that can be enabled or disabled.	Value <input type="checkbox"/> Enable
Word	An option that holds an integer value. It also has a minimum and a maximum. If an invalid value is entered, a message will tell you that, together with the allowed range of values.	Value <input type="text" value="23"/>
String	An option that holds a text value.	Value <input type="text" value="192.168.1.231"/>
List	An option for which you can choose from a list of values.	Value <div> <input type="text" value="IBM-3278-2"/> <div> <div></div> <div>VT52/100/220</div> <div>IBM-3278-2</div> <div>IBM-3279-2</div> </div> </div>
Color	A color option. Click on the button to select a color from the color palette dialog box.	Value <input type="color" value="blue"/>
Color Pair	A color option that sets both foreground and background colors. There is also an area that display a sample with your color selection.	Value <div> <div>Foreground</div> <div>Background</div> <div>SAMPLE</div> </div>



## The CEInstaller Application

CEInstaller is started when you click the install button .

It transfers the configuration, the emulator, some support files, and any scripts you might have written onto the CE device.



Above you see a typical installation.  
CEInstaller connects to the CE device through ActiveSync.

If you already have installed the emulator and just wants to download a new configuration, select the *Configuration* in the Files listbox.

During the installation process the application folder is created and files are transferred.

If the emulator has been previously installed and is running on the device, you get the error message *Failed to create file ... on CE device. Maybe the emulator is running?*. Exit the emulator on the device and retry the installation. If the problem persists, reboot the CE device, preferably with a hardware reset button.

The emulator is installed in Flash memory (on selected devices). Even if the unit runs out of batteries or is cold-booted you don't need to re-install the emulator. You might have to re-configure the RF card though.

### ***What is transferred during installation and where is it?***

The CEInstaller installs the files from the directory on your PC that corresponds to the machine selected in the Configuration Application.

For example, the files installed to a Denso BHT-200 are located in

*C:\Program Files\Freefloat\Freefloat Access One\Denso\BHT200*

As an example, let's add a script to the folder for the BHT-200.

1. Start Notepad.
2. Enter the following text:

```
Label loopx  
SendKey <pf1>  
SendKey <pf2>  
SendKey <pf3>  
SendKey <pf4>  
Goto loopx
```

3. Save it as Test.aut in the folder  
*C:\Program Files\Freefloat\Freefloat Access One\Denso\BHT200*.  
(The path might be different depending on where you installed the product on your PC.)
4. Start CEInstaller and select *Scripts* in the listbox Files. Then click on the Install button.
5. As you can see the script file you just created is transferred to the CE device.

## Keyboard Mapping

Keys can be configured to generate text data and/or simulate other keys. Each emulation has its own definition file containing the keyboard mappings.

VT/ANSI's definition are in the file *ansikeymap.txt*, 3270's are in *3270keymap.txt*, and 5250's are in *5250keymap.txt*.

A key can also be mapped to start a script.

Key mappings can be dynamically changed in a script using the script verb **SetKey**.

If line mode buffering is used in VT/ANSI, please note that all characters generated from a key mapping are not sent directly to host. Append `<flush>` to send the characters in the key mapping directly to the host. For example:

```
KEYF5="\x1bOT<flush>"
```

### How to map a key

1. Open the configuration file for the device you are using. For example: Select *Config Denso BHT-200* from the Start menu.
2. Select the **Edit VT Keymap**, **Edit 3270 Keymap**, or **Edit 5250 Keymap** command in the configuration program depending on what emulation you intend to use.
3. This opens the correct keymap file in a text editor (normally Notepad).
4. Let's say you want to configure F1 to send the text "HELLO" and then press the Enter key.
5. In the text editor go to the line containing KEYF1, or if you can't find it, add a line looking like this at the end of the keymap file:

```
KEYF1="HELLO<enter>"
```

KEYF1 is the key identifier that identifies the hardware key. Inside a key mapping keys are denoted by enclosing its name in angular brackets `<>`. For all key identifiers and key names, please see **Key Table** below.

In 3270 and 5250 the key name (`<enter>`) means the action performed by the emulation when that key is pressed. That is, if KEYENTER is mapped to generate the text "hello" then `<enter>` in another key mapping still performs the actual Enter action.

Note that a key mapping can contain a sequence of text and/or multiple keys, for example: `KEYF5="USER<tab>PASSWORD<enter>"`.

## How to start a script from a key mapping

Add a key mapping using the following syntax:

```
KEYF2="<script:test.aut>"
```

The above key mapping starts the script `test.aut`. Please note that any keys or text after the instruction to start a script is ignored.

## Key Table

Key Identifier	Key Name	Key Name Alias	Comments
KEYCTRLA	<ctrl-a>		
KEYCTRLB	<ctrl-b>		
KEYCTRLC	<ctrl-c>		
KEYCTRLD	<ctrl-d>		
KEYCTRLE	<ctrl-e>		
KEYCTRLF	<ctrl-f>		
KEYCTRLG	<ctrl-g>		
KEYCTRLH	<ctrl-h>		
KEYCTRLI	<ctrl-i>		
KEYCTRLJ	<ctrl-j>		
KEYCTRLK	<ctrl-k>		
KEYCTRLL	<ctrl-l>		
KEYCTRLM	<ctrl-m>		
KEYCTRLN	<ctrl-n>		
KEYCTRL O	<ctrl-o>		
KEYCTRLP	<ctrl-p>		
KEYCTRLQ	<ctrl-q>		
KEYCTRLR	<ctrl-r>		
KEYCTRLS	<ctrl-s>		
KEYCTRLT	<ctrl-t>		
KEYCTRLU	<ctrl-u>		
KEYCTRLV	<ctrl-v>		
KEYCTRLW	<ctrl-w>		
KEYCTRLX	<ctrl-x>		
KEYCTRL Y	<ctrl-y>		
KEYCTRLZ	<ctrl-z>		
KEYTAB	<tab>		
KEYBACKTAB	<backtab>		
KEYESC	<escape>	<esc>	
KEYENTER	<enter>		

KEYBACKSPACE	<backspace>		
KEYNEWLINE	<newline>		5250 Newline
KEYF1	<pf1>	<f1>	
KEYF2	<pf2>	<f2>	
KEYF3	<pf3>	<f3>	
KEYF4	<pf4>	<f4>	
KEYF5	<pf5>	<f5>	
KEYF6	<pf6>	<f6>	
KEYF7	<pf7>	<f7>	
KEYF8	<pf8>	<f8>	
KEYF9	<pf9>	<f9>	
KEYF10	<pf10>	<f10>	
KEYF11	<pf11>	<f11>	
KEYF12	<pf12>	<f12>	
KEYF13	<pf13>	<f13>	
KEYF14	<pf14>	<f14>	
KEYF15	<pf15>	<f15>	
KEYF16	<pf16>	<f16>	
KEYF17	<pf17>	<f17>	
KEYF18	<pf18>	<f18>	
KEYF19	<pf19>	<f19>	
KEYF20	<pf20>	<f20>	
KEYF21	<pf21>	<f21>	
KEYF22	<pf22>	<f22>	
KEYF23	<pf23>	<f23>	
KEYF24	<pf24>	<f24>	
KEYLEFT	<left>		
KEYRIGHT	<right>		
KEYUP	<up>		
KEYDOWN	<down>		
KEYSHIFLEFT	<shift-left>		
KEYSHIFTRIGHT	<shift-right>		
KEYSHIFTUP	<shift-up>		
KEYSHIFTDOWN	<shift-down>		
KEYALTLEFT	<alt-left>		Scrolls the viewport.
KEYALTRIGHT	<alt-right>		Scrolls the viewport.
KEYALTUP	<alt-up>		Scrolls the viewport.
KEYALTDOWN	<alt-down>		Scrolls the viewport.
KEYINS	<insert>		
KEYDEL	<delete>		
KEYEND	<end>		
KEYHOME	<home>		
KEYPGDN	<pagedown>		
KEYPGUP	<pageup>		

KEYNUMDIVIDE	<numdivide>		
KEYNUMMULTIPLY	<nummultiply>		
KEYNUMSUBTRACT	<numsubtract>		
KEYNUMADD	<numadd>		
KEYCLEARFIELD	<clearfield>		Clears the current field in 3270 and 5250.
KEYNUMENTER	<numenter>		
KEYALTDEL	<alt-delete>		3270 Delete Word
KEYALTP	<alt-p>		Toggles the keyboard (Key*One).
KEYCTRLPGUP	<ctrl-pageup>		
KEYALTNUMADD	<alt-numadd>		Increases the font height.
KEYALTNUMSUBTRACT	<alt-numsubtract>		Decreases the font height.
KEYCTRLNUMADD	<ctrl-numadd>		Increases the font width.
KEYCTRLNUMSUBTRACT	<ctrl-numsubtract>		Decreases the font width.
KEYATTN	<attention>		
KEYSYSREQ	<sysreq>		
KEYCLEAR	<clear>		
KEYDUP	<dup>		
KEYERASE	<erase>		
KEYFIELDEXIT	<fieldexit>		
KEYFIELDMINUS	<fieldminus>		
KEYFIELDPLUS	<fieldplus>		
KEYHELP	<help>		
KEYPRINT	<print>		
KEYPA1	<pa1>		
KEYPA2	<pa2>		
KEYPA3	<pa3>		
KEYM1	<m1>		M1,M2 and M3 are special keys on Denso BHT handhelds.
KEYM2	<m2>		
KEYM3	<m3>		
KEYSHIFTM1	<shift-m1>		
KEYSHIFTM2	<shift-m2>		
KEYSHIFTM3	<shift-m3>		
KEYSEND	<send>		Special key on Denso BHT 400.
KEYFLUSH	<flush>		Flushes the VT/ANSI line mode buffer.
KEYTERMINATE	<terminate>		Terminate/close session

KEYRESTART	<restart>		Disconnect/connect session
KEYREBOOT	<reboot>		Reboots the PC. Reboot type controlled by configuration setting <i>Win32/Forced Reboot</i> .
<i>The following keys have no mapping by default:</i>			
KEYPAUSE	<pause>		
KEYCTRLBREAK	<ctrl-break>		
KEYSHIFTESC	<shift-esc>		
KEYLSHIFT	<left-shift>		
KEYRSHIFT	<right-shift>		
KEYLCTRL	<left-ctrl>		
KEYRCTRL	<right-ctrl>		
KEYLALT	<left-alt>		
KEYRALT	<right-alt>		Note! The Right Alt key and the AltGr key are the same physical key but it uses different codes depending on your keyboard language.
KEYALTGR	<altgr>		
KEYAPPS	<apps>		

## Configurable Keyboard

The on-screen keyboard is configurable. To modify the default keyboard layouts or to create a new keyboard layout, please download Freefloat Key\*One and use the Designer application. You do not need a license for Key\*One to be able to use the Designer.

Please refer to the Key\*One documentation on how to use the Designer.



## Scripting

Scripting is a means to automate things, such as logging on to a host, or navigate a menu system to the desired user application etc. It can also be used to change the behavior of the emulator.

A script is a text file (single byte, not Unicode) containing instructions that the emulator obeys.

For a script to be selectable from the script dialog box, it **must** have a file extension of **.aut**.

### Hotspot Scripts

Note that hotspot scripts are only active when the parameter *Mouse/Touch Mode* is set to **Hotspot**.

The only differences between a hotspot script and a normal script is that hotspot scripts have the file extension **.hot** and that hotspot scripts can only be started by clicking on a word in the emulator window.

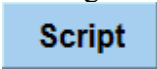
If you, for example, clicks on the word `Cancel` in the emulator window, the emulator will open and run the hotspot script **cancel.hot**. If there is no hotspot script named **cancel.hot**, nothing will happen and an error is logged in the log file.

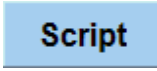
When you click at a word in the emulator, the emulator scans left and right from the position you clicked. All letters (including international letters) and digits are included in the hotspot word. Other characters, like hyphen or slash, will terminate the hotspot word scan. For example, if you click on the letter y in the word `IBM-system` the hotspot word will be `system` and not `IBM-system`.

### ***Starting and Stopping Scripts***

A script can be started in three ways, either automatically on starting the emulator through the *Start Script* option, or selected from the script dialog box at any time during a host session, or via a mapped key.

If a script is automatically started on startup, it begins executing as soon as the session has been established.

To select a script for execution, or checking the status of a running script, bring up the script dialog box by clicking on the  button on the keyboard.

If a script is running when you click on the  button, the emulator asks you if you want to stop it.

When you select a script in the script dialog box and click on OK, the selected script is started.

## Script Commands

Verb	Type1	Type2	Type3	Type4
WaitForChange				
WaitForCursor				
WaitForAny				
Sleep	Seconds			
Label	Text			
Goto	Text			
SendString	Text			
SendKey	Key Name			
SetKey	Text			
WaitForText	Row	Column	Text	
WaitWhileText	Row	Column	Text	
CheckForText	Row	Column	Text	Target Label
ScanForText	Text	Target Label		
LockWindow				
UnlockWindow				
SetOrigin	Row	Column		
ChainScript	Text			
SetParameter	Text			
ErrorAlert	Row	Column	Text	
WaitCursorAt	Row	Column		
WaitCursorAbove	Row	Column		
WaitCursorBelow	Row	Column		
WhenCursorAt	Row	Column	Target Label	
WhenCursorAbove	Row	Column	Target Label	
WhenCursorBelow	Row	Column	Target Label	
SetCursorPos	Row	Column		
FindCursor				
EnableScanner				
DisableScanner				
FontSize	Height	Width		
Terminate				
PlaySound	Text			
ShowKeyboard				
HideKeyboard				
ShowCanvas	Canvas Number			
HideCanvas	Canvas Number			
SetCanvasPosition	Canvas Number	X Pos	Y Pos	
SnapCanvas	Canvas Number	Snap Position		
ShowAllCanvases				
HideAllCanvases				

### WaitForChange

Suspends the script execution until any text changes on the screen. Execution resumes at the next script-line as soon as any text is changed on the screen.

### WaitForCursor

Suspends the script execution until cursor moves. Execution resumes at the next script-line as soon as cursor position has changed.

### WaitForAny

Suspends the script execution until either some text changes on the screen or the cursor is moved. Execution resumes at the next script-line as soon as either action happens.

**Sleep *n***

Delays the script execution for *n* seconds. Valid values for *n* are 1 to 99.

**Label *Targetname***

Defines a branch target name used by the verbs CheckForText, ScanForText and Goto.

**Goto *Targetname***

Script execution resumes unconditionally at the label *Targetname*.

**SendString "*String*"**

Sends the characters in *String* as a stream of keyboard characters to the host.

Use the command **SendKey** (se below) to terminate the string

**SendKey <*Key Name*>**

Executes the action associated with the key. See the section **Key Table** for the valid key names.

**SetKey <*Key Definition String*>**

Assigns a keymapping.

```
SetKey KEYF1="abc "
```

The above maps the F1 key to generate the text abc. See the section **Key Mapping** for information.

**WaitForText *Row, Col, "Text"***

Suspends the script execution until the defined text appears at the defined coordinate.

Execution resumes at the next script-line as soon as the defined text appears at the defined coordinate.

**WaitWhileText *Row, Col, "Text"***

Suspends the script execution until the defined text disappears from the defined coordinate.

Execution resumes at the next script-line as soon as the defined text disappears from the defined coordinate

**CheckForText *Row, Col, "Text", Target Label***

If the specified text is present at the defined coordinate, script execution resumes at the target label.

**ScanForText "*Text*", Target Label**

If the specified text is present anywhere on the screen, script execution resumes at the target label.

### **SetOrigin *Row, Col***

Defines the position of the host screen that corresponds to the visible screens upper left corner.

### **LockWindow & UnlockWindow**

Locks or unlocks the manual scrolling capability of the terminal.

While the screen is locked, it is not possible for the operator to move the screen viewport manually, only the script command SetOrigin can move the viewport.

### **ChainScript "*FileName*"**

Terminates the current script and begins execution of a new script file.

### **SetParameter "*String*"**

This command can change any of the internal settings normally controlled by the configuration file Terminal.cfg, at any time. The "*String*" is any valid Terminal.cfg command string.

Example: assuming the terminal is configured to send a TAB after any barcode scanned, the option *Barcode Suffix* is set to <tab>.

However, at some point it is desired to change that to sending a ENTER instead, the following command in the script would do that:

```
SetParameter "Common\Barcode\Barcode Suffix=<enter>"
```

The path and name of each setting can be found by opening the configuration file in a text editor.

### **ErrorAlert *Row, Col, "Text"***

A special command for 3270 or 5250 mode only.

These terminals normally only has one kind of audible beep.

When using a portable scanner in industrial environment it is often desirable to have audible feedback both on good reads and when some kind of error occur.

The ErrorAlert command enables us to use the normal screen attribute "Sound alarm" as a "good-read" indicator, and simultaneously define an alternate audible signal used for errors.

When a new screen arrives, and the user defined "Text" is present at the specified coordinates (Row, Col), the terminal will sound the *Error Tone\Frequency* for duration *Error Tone\Time* (see the configuration program for a detailed explanation of this parameter.)

To disable it, just set the “*Text*” to some text that cannot appear at the coordinates.

Since this is a script command, the text and coordinates can be dynamically changed whenever needed.

**WaitCursorAbove *Row, Col***

**WaitCursorAt *Row, Col***

**WaitCursorBelow *Row, Col***

Suspends the script execution until the cursor position is above/at/below the defined coordinate.

Execution resumes at the next script-line as soon as the designated condition is true.

**WhenCursorAbove *Row, Col, Target Label***

**WhenCursorAt *Row, Col, Target Label***

**WhenCursorBelow *Row, Col, Target Label***

If the cursor position is above/at/below the defined coordinate, script execution resumes at the target label.

Jumps (branches) to the target label if the designated condition is true.

**SetCursorPos *Row, Col***

Only available in 3270 and 5250 mode.

Moves the cursor (active insertion point) to the specified coordinates.

**FindCursor**

Locate the cursor and center the visible screen around it. The same as clicking on the *Find Cursor Area*.

**EnableScanner**

Enables the barcode reader. Note that in 5250 and 3270 emulation the cursor also has to be in an input field before scanning is possible.

**DisableScanner**

Disables the barcode reader. The scanner remains disabled until the script command EnableScanner is executed. This is one way to turn off scanning in ANSI/VT emulation. The other is to control the scanner from the host program via escape sequences.

*Note!* This also turns off barcode processing when a external wedge program is used (if the wedge sends barcode data by the Ctrl+V key).

**Terminate**

Terminates the host session, disconnects from the host.

**PlaySound "Sound name"**

Plays the sound file specified (it must be in .wav format). The special sound names "Normal", "Barcode", and "Error" plays the built-in wave files, normal.wav, barcode.wav and error.wav. For these sounds the settings Read Confirmation, Bell, and Error Tone also affects how and if the sound is played.

**ShowKeyboard**

Shows Key\*One.

**HideKeyboard**

Hides Key\*One.

**ShowCanvas Canvas Number**

Show the specified canvas of Key\*One.

**HideCanvas Canvas Number**

Hides the specified canvas of Key\*One.

**SetCanvasPosition Canvas Number, X Pos, Y Pos**

Moves the specified canvas of Key\*One to a new position. The X Pos and Y Pos are screen coordinates.

**SnapCanvas Canvas Number, Snap Position**

Moves the specified canvas of Key\*One to a snap position.

Snap Position	Canvas Moved To
0	Upper Left
1	Upper Middle
2	Upper Right
3	Lower Left
4	Lower Middle
5	Lower Right

**ShowAllCanvases**

Shows all canvases of Key\*One.

**HideAllCanvases**

Hides all canvases of Key\*One.

## Appendix A – Data Transmitted by Various Keys in VT/ANSI Mode

### Function Keys

Key	ANSI Characters Generated
PF1 / F1	Esc O P
PF2 / F2	Esc O Q
PF3 / F3	Esc O R
PF4 / F4	Esc O S
F5	Esc O T
F6	Esc [ 1 7 ~
F7	Esc [ 1 8 ~
F8	Esc [ 1 9 ~
F9	Esc [ 2 0 ~
F10	Esc [ 2 1 ~
F11	Esc [ 2 3 ~
F12	Esc [ 2 4 ~
F13	Esc [ 2 5 ~
F14	Esc [ 2 6 ~
F15	Esc [ 2 8 ~
F16	Esc [ 2 9 ~
F17	Esc [ 3 1 ~
F18	Esc [ 3 2 ~
F19	Esc [ 3 3 ~
F20	Esc [ 3 4 ~

### Cursor Keys

Cursor	ANSI Characters Generated	
	Reset (Cursor)	Set (Application)
Up	ESC [ A	ESC O A
Down	ESC [ B	ESC O B
Right	ESC [ C	ESC O C
Left	ESC [ D	ESC O D

Please note that all strings and codes that are sent when keys are pressed in VT/ANSI mode are configurable. Please see the text file *ansikeymap.txt* for more information.

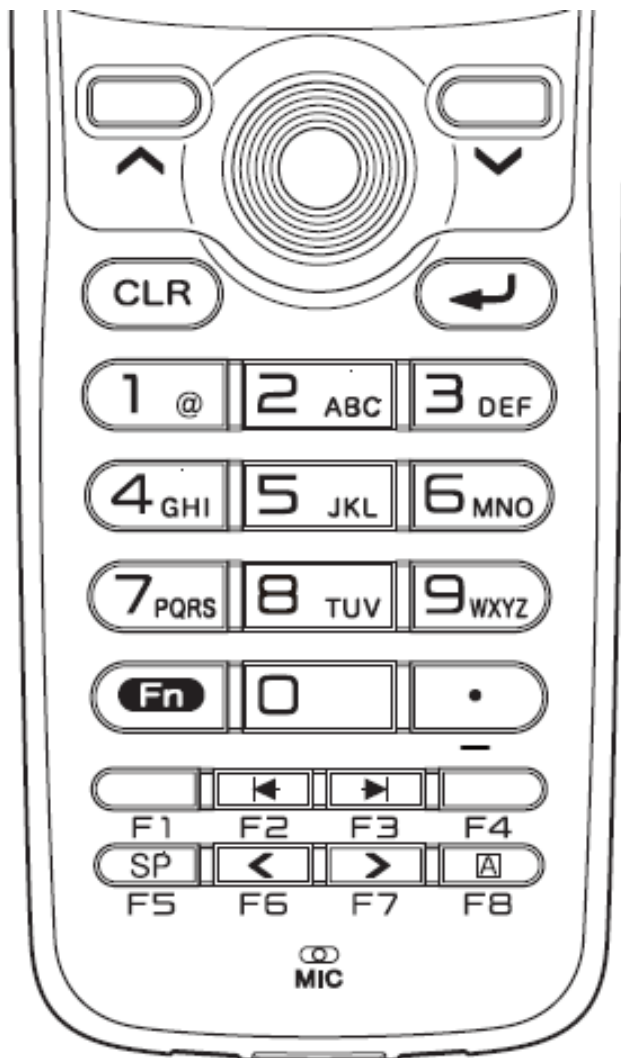


## Appendix B – Model Specifics

### Casio DT-X7

#### Keyboard

The Casio DT-X7 has no touch screen. To be able to navigate and use terminal specific keys, two extra keyboard modes has been added. Note that these modes are in addition to the standard modes of the keyboard (numeric, lower case, and uppercase). The standard modes are changed using the **A** key.



The two additional modes are called **Function key mode 1 (f1)** and **Function key mode 2 (f2)**. The status bar will indicate which one of the five possible modes which is active.

You toggle between one of the standard modes and the extra keyboard modes by pressing the yellow key (F4). All keys but arrow keys are "single shot" keys. It means that the keyboard mode will go back to the standard mode after pressing Yellow and another key.

You can also cycle through the standard mode and the extra modes by pressing the Yellow key repeatedly.

### Function Key Mode 1

Key Sequence	Action	Stays in mode
Yellow + CLR	Reset/unlock (5250/3270)	
Yellow + Enter	Centers the display around the cursor	
Yellow + 1 to 9	Sends F1 to F9 to the host	
Yellow + 0	Sends F10 to the host	
Yellow + up arrow	Scroll 1 line up	Yes
Yellow + down arrow	Scroll 1 line down	Yes
Yellow + left arrow	Scroll 1 column left	Yes
Yellow + right arrow	Scroll 1 column right	Yes

### Function Key Mode 2

Key Sequence	Action	Stays in mode
Yellow + Yellow + Enter	Centers the display around the cursor	
Yellow + Yellow + 1 to 9	Sends F11 to F19 to the host	
Yellow + Yellow + 0	Sends F20 to the host	
Yellow + Yellow + up arrow	Increase font height	Yes
Yellow + Yellow + down arrow	Decrease font height	Yes
Yellow + Yellow + left arrow	Decrease font width	Yes
Yellow + Yellow + right arrow	Increase font height	Yes

**Note!** Red + CLR disconnects the session (a confirmation dialog will be displayed).

### Denso BHT-200

#### Keyboard

The M1 key toggles the keyboard. The M2 key works as the TAB key on a normal keyboard. Shift + M2 is the same as Shift + TAB.

On the BHT-200 with alphanumeric keyboard, you can activate the alpha mode by pressing and holding down the SF/ALP key until you hear a beep. The physical keyboard is then toggled between numeric and alpha input modes. The currently active mode is indicated by a N or an A in the status bar.

T	3270	N	1/1
---	------	---	-----

If you need quick access to the function keys F11 to F20 you can press and hold the SF (shift) key and then a function key. F1 is then F11, F2 is F12 and so on. Please note that this only works in numeric input mode.

You can save the RF setup by holding down the SF key and then powering off the unit. The RF setup will then survive even if the battery is completely drained.

#### Pasting

When running Access\*One on a BHT-200 together with kbifCE (supplied from Denso Wave) there is a config option that can speed up barcode handling: *Use Ctrl-V as paste command*. When enabled the key Ctrl-V is used as the paste command. When disabled Ctrl-V is used as a normal key, which in ANSI/VT emulation is treated as a control character.

Please note that you also need to choose the Send Mode *Use Clipboard* in kbifCE.

Using the paste method to transfer barcode data from kbifCE to Access\*One is approximately 45 times faster than using the kbifCE Send Mode *Keyboard Emulation*.

When kbifCE is not used, Access\*One has control over the barcode reader, and in this case the option has no effect.

### Denso BHT-400

On a Denso BHT-400, Access\*One has the same indicator for numeric and alpha input modes as BHT-200. Please see more information above.

In addition, on BHT-400, an F is displayed when the keyboard function mode has been activated.

### Symbol PPT8846

Please note that only the 15-key variant of the PPT8846 is supported.

The emulator remaps FUNC+keys to enable use of function keys. This remapping overrides any functions mapped in the operating system. The normal mappings are restored when you exit the emulator.

### Machines that have a sound device capable of playing wav-files

In the folders for those devices there are some wave files. Feel free to replace them with your favorite sounds.

Filename	Event
startup.wav	When the emulator is started
normal.wav	When a new form with the sound alarm bit set is received in 3270/5250 emulation or when the bell characters is received in VT/ANSI emulation
barcode.wav	When a barcode is read
error.wav	When the keyboard is locked in 3270 or 5250 emulation

### PSC Falcon 44xx with monochrome screen

The configuration for the PSC Falcon 44xx is intended for the color model. If you have a monochrome 44xx, insert these statements into the configuration file to get a black and white color scheme that is a bit more readable than the gray scale.

```
// Color Options
3270 Emulation\Colors\Default Background=16777215
5250 Emulation\Colors\Default Foreground=0
5250 Emulation\Colors\Default Background=16777215
VT/ANSI Emulation\Colors\Black=16777215
VT/ANSI Emulation\Colors\Blue=0
VT/ANSI Emulation\Colors\Green=0
VT/ANSI Emulation\Colors\Cyan=0
VT/ANSI Emulation\Colors\Red=0
VT/ANSI Emulation\Colors\Magenta=0
VT/ANSI Emulation\Colors\Brown=0
```

```
VT/ANSI Emulation\Colors\Light Gray=0
VT/ANSI Emulation\Colors\Dark Gray=0
VT/ANSI Emulation\Colors\Light Blue=0
VT/ANSI Emulation\Colors\Light Green=0
VT/ANSI Emulation\Colors\Light Cyan=0
VT/ANSI Emulation\Colors\Light Red=0
VT/ANSI Emulation\Colors\Light Magenta=0
VT/ANSI Emulation\Colors\Yellow=0
VT/ANSI Emulation\Colors\White=0

// Color Pair Options
3270 Emulation\Colors\Basic Colors\Unprotected Normal=0,16777215
3270 Emulation\Colors\Basic Colors\Protected Normal=0,16777215
3270 Emulation\Colors\Basic Colors\Unprotected Intensified=0,16777215
3270 Emulation\Colors\Basic Colors\Protected Intensified=0,16777215
5250 Emulation\Colors\Default Intensified=0,16777215
5250 Emulation\Colors\Green=0,16777215
5250 Emulation\Colors\White=0,16777215
5250 Emulation\Colors\Red=0,16777215
5250 Emulation\Colors\Turquoise=0,16777215
5250 Emulation\Colors\Yellow=0,16777215
5250 Emulation\Colors\Pink=0,16777215
5250 Emulation\Colors\Blue=0,16777215
3270 Emulation\Colors\Extended Colors\Blue=0,16777215
3270 Emulation\Colors\Extended Colors\Green=0,16777215
3270 Emulation\Colors\Extended Colors\Pink=0,16777215
3270 Emulation\Colors\Extended Colors\Red=0,16777215
3270 Emulation\Colors\Extended Colors\Turquoise=0,16777215
3270 Emulation\Colors\Extended Colors\White=0,16777215
3270 Emulation\Colors\Extended Colors\Yellow=0,16777215
3270 Emulation\Colors\Extended Colors\Default Normal=0,16777215
3270 Emulation\Colors\Extended Colors\Default Intensified=0,16777215
```

## Appendix C – Logging Network Data

When the option *Raw Logging* is turned on, the emulator stores all network data into a file called `rawlog.txt`.

The log file can be used by an engineer for diagnostics.

Please note that raw logging can only be used when running Freefloat Access One in regular Windows.

Raw Logging can be turned on and off via the Log menu. It can also be cleared and the formatted log file can be viewed from the menu.

## Appendix D – EBCDIC Character Tables

The default code page used in the Freefloat Emulator is 278. Please see topic Display Character Translation on how to modify the translation tables.

### Code Page 278, Swedish/Finnish

HEX DIGITS	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
1ST → 2ND ↓												
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	μ SM170000	¢ SC040000	ä LA170000	å LA270000	É LE120000	0 ND100000
-1	(RSP) SP300000	` SD130000	/ SP120000	\ SM070000	a LA010000	j LJ010000	ü LU170000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	{ SM110000	ë LE170000	# SM010000	Ë LE180000	c LC010000	l LL010000	t LT010000	• SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	[ SM060000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	} SM140000	ï LI170000	\$ SC030000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ï LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	é LE110000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	§ SM240000	☒ SC010000	ö LO170000	:	« SP170000	⌘ SM210000	ï SP030000	¬ SM660000	(SHY) SP320000	1 ND011000	2 ND021000	3 ND031000
-B	• SP110000	Ä LA280000	, SP080000	Ä LA180000	» SP180000	º SM200000	¿ SP160000	 SM130000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	Ö LO180000	ð LD630000	æ LA510000	Ð LD620000	- SM150000	! SM650000	~ SD190000	@ SM050000	Ü LU180000
-D	( SP060000	) SP070000	— SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	' SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	± SA020000	J SM080000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00278

# Code Page 37, USA/Canada

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	μ SM170000	^ SD150000	{ SM110000	}	\ SM070000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	Ê LE120000	a LA010000	j LJ010000	~ SD190000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	• SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ï LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	` SD130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	¢ SC040000	! SP020000	¡ SM650000	: SP130000	« SP170000	ª SM210000	ï SP030000	[ SM060000	(SHY) SP320000	1 ND011000	2 ND021000	3 ND031000
-B	· SP110000	\$ SC030000	, SP080000	# SM010000	» SP180000	º SM200000	¿ SP160000	]	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	@ SM050000	ð LD630000	æ LA510000	Ð LD620000	- SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	( SP060000	) SP070000	— SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	' SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	 SM130000	¬ SM660000	? SP150000	" SP040000	± SA020000	☒ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00037

IBM's codepage documentation can be found here:

<http://www-03.ibm.com/systems/i/software/globalization/codepages.html>



## Appendix E – TN5250 Auto-signon

Telnet supports the capability for a user to bypass the Sign On display by sending a user profile name and password with the Telnet session request. The 5250 host uses the setting for the QRMTSIGN (Remote sign-on) system value to determine how to handle requests for automatic sign-on. The table below shows the options. These options apply only when the Telnet request includes a user ID and password.

The password is sent in an encrypted form according to the document RFC2877 (5250 Telnet Enhancements).

How QRMTSIGN Works with Telnet	
*REJECT	Telnet sessions that request automatic sign-on are not allowed.
*VERIFY	If the user profile and password combination is valid, the Telnet session starts.
*SAMEPRF	If the user profile and password combination is valid, the Telnet session starts.
*FRCSIGNON	The system ignores the user profile and password. The user sees the Sign-On display.

You set the username and password in the configuration program. The parameters are:

```
5250 Emulation\TN5250 Enhancements\User Name
5250 Emulation\TN5250 Enhancements>Password
```

## Appendix F – TN5250 Enhancement Settings

The setting for TN5250 Enhancements can be specified in an external file. This is especially useful when those parameters are different between individual terminals. For example, some installations use a unique device name for each terminal to be able to identify the sessions.

Put these settings in a text file named TN5250Enhancements.txt in the same folder as your configuration file. You can specify one or more of the settings in the following format:

```
User Name=<user name>
Password=<password>
Current Library=<library>
Initial Menu=<menu>
Program To Call=<program>
Device Name=<device name>
```

## Appendix G – Command Line Parameters

Command line parameters can be passed to Access\*One when started from a shortcut or registry key.

Currently, two parameters are supported:

`[-autostart] [configuration file]`

The *autostart* parameter, when specified, activates the *Startup Delay Time* option in the configuration file. The delay can be used when Access\*One is started before the network connection is initialized.

For example, a possible command line for starting Access\*One could look like this:

```
"C:\Program Files\Freefloat\Freefloat Access One\Win32\win32.exe -autostart  
C:\Program Files\Freefloat\Freefloat Access One\Win32\terminal.cfg"
```

## Appendix H – WLinQ Bell integration (WIN32 only)

The WLinQ Bell option monitors texts on the terminal emulator screen and when they appear data are sent back to the unit connected to WLinQ, for example a scanner.

In most cases, WLinQ Bell is used for adding error feedback by sending a beep to a wireless scanner attached to the terminal thus alerting the operator that there is some error condition requiring attention, i.e. go back to the terminal and read the error message.

WLinq Bell option requires Freefloat WLinQ 3.14 or later.

### **Configuration**

The configuration file contains the texts that should be monitored with each text definition on a separate line. The name of the configuration file is `WLinqBell.ini`. If the configuration file is present, the WLinQ Bell integration option is activated.

Each text definition is a comma separated list of values:

*Trigger Text, Text Row, Text Column, Cursor Row, Cursor Column, Data File*

### **Trigger Text**

The text to trigger on.

### **Text Row/Text Column**

The row/column where *Trigger Text* will appear on the terminal emulator's screen.

### **Cursor Row/Cursor Column**

The location of the cursor on the terminal emulator's screen determining when the *Data File* connected to the text should be sent to WLinQ.

This is needed to avoid generating multiple events when the screen is modified locally by the user, i.e. moving the cursor around and/or entering data in a field.

### **Data File**

When *Trigger Text* appear at (*Text Row, Text Column*) and the cursor is positioned at (*Cursor Row, Cursor Column*) the contents of the file named *Data File* is sent to WLinQ.

### **Sample Configuration**

```
// <trigger text>,<text row>,<text column>,  
// <cursor row>,<cursor column>,<data file>  
73105541,5,2,5,10,BEEP.BIN
```

The above sample configuration will make WLink Bell monitor for the text 73105541 appearing at row 5, column 2 and the cursor positioned at column 5, row 10. When this happens, the content of the file BEEP.BIN is sent to WLink.

The sample file BEEP.BIN contains the data <bel><cr> (ASCII 7, ASCII 13) which will make certain equipment like barcode scanners beep.

You may need to create a suitable data file for the equipment connected to WLink.

## Appendix I – Version History

Version	Changes	Date
1.3.0	<p>Changes to license check. If a valid license for some reason is reported as incorrect, a log line is created with all data needed for debugging.</p> <p>This version accepts both version 1.1 and 1.2 license keys, no need to convert license when upgrading version.</p> <p>Committing a character in alpha mode with the ENT key on a Denso BHT-200 now works.</p>	2005-03-07
1.3.1	Demo time was too short.	2005-03-09
1.3.2	<p>Added support for printing in ANSI mode.</p> <p>Re-introduced COURH.TTF for non-WIN32 platforms, needed for the ANSI linedrawing characters on CE</p>	2005-03-21
1.3.3	<p>Changes to license check. A license key for version 1.1 or version 1.2 can be in either WINCEEMU.LIC or WINCE120.LIC, if the license key is valid it no longer matters which of the two filenames is used to store the key.</p> <p>Fixed a bug in ANSI where text could sometimes be rendered invisible if new text was printed to a position which had the blink attribute set.</p>	2005-04-14
1.3.4	<p>Keystrokes can now be encoded in barcodes. <b>Note!</b> This only applies to models where Freefloat Access*One has direct control of the barcode device, i.e. Denso BHT-200 models and Symbol PPT8846 models.</p> <p>The popup toggle, when mapped to a key, did not work for hiding the popup. This only applied to Windows CE versions. Fixed.</p>	2005-05-10
1.3.5	<p>Removed the binary version of the network data log. It proved to be unnecessary.</p> <p>If the RF card couldn't associate with an AP, Access*One no longer exits. This was a problem, because it could give the user access to the CE shell even if an exit password was set. Only applicable to Denso BHT-200.</p> <p>The support for the Denso BHT-100 has been dropped.</p>	2005-05-13
1.3.6	Enabled add-on codes in Denso BHT-200 for EAN-13, UPC-A, EAN 8 and UPC-E.	2005-05-16

Version	Changes	Date
1.3.7	Added user-configurable Primary DA response string (VT/ANSI).	2005-08-11
1.3.8	Added the configuration option Use Ctrl-V as paste command. The new topic Pasting explains this option.	2005-08-31
1.3.9	Updated 5250 Test Server so it can use another IP port (port passed on the command line)	2005-09-02
1.3.10	Minor fix to 5250 Test Server, Telnet negotiation problem with 5250 environment variables.  Minor fix to popup snapping at negative coordinates.	2005-10-13
1.3.11	Added support for the Intermec CN2 device.	2005-10-14
1.3.12	The popup keyboard window is now created hidden. This avoids some flicker when Access*One is started.  Changed layout of Unregistered and Register dialogs and the Splash window so all important buttons and controls are usable on a QVGA in landscape mode.  Tighter integration of the Intermec CN2 device to make barcode scanning much faster.  The log file is trimmed to the last 100 lines.	2005-10-21
1.3.13	Added the script keywords EnableScanner and DisableScanner.  Scripts are now installed in "\My Documents" on Pocket PC units. This is because Pocket PC can't browse for files anywhere else. The browse function is used when starting a script manually.  Added the script command PlaySound.	2005-10-24
1.3.14	Added the script command SetCanvas.	2005-11-11
1.3.15	Added support for the DLoG X7 device. The external Enter key was a numeric Enter. Translations added in the keymap files for each emulation type.	2005-11-29
1.3.16	Added support for the DLoG XMDA device. Added support for the Denso BHT-400 device.	2005-12-23
1.3.17	Added support for the Symbol MC3070 device	2006-01-10
1.3.18	Fixed a bug in 5250 error message handling	2006-01-12
1.3.19	Fixed bugs in ANSI introduced in 1.3.7	2006-01-27
1.3.20	Added feature: popup keyboard buttons can now contain strings.  Removed limitation of max 20 characters on a key's output in popup designer.	2006-02-02

Version	Changes	Date
1.3.21	Added the script commands ShowPopup and HidePopup	2006-02-02
1.3.22	Customized project version	2006-02-10
1.3.23	<p>Removed the special handling of script location introduced in version 1.3.13. Only affects Pocket PC devices.</p> <p>Added support for detection of network AP attachment for Symbol Radio's.</p> <p>Popup keyboard is now better at handling differently sized canvases.</p> <p>An option in CEInstaller has been added so only files related to the popup keyboard can be sent to a device.</p>	2006-02-15
1.3.24	Bugfix for Customized project version.	2006-02-20
1.3.25	Fixed potential bug in popup keyboard. Increased limit of possible font sizes to 50.	2006-02-20
1.3.26	Fix for Denso BHT-400, wait until the radio is fully initialized before trying to connect to host.	2006-03-02
1.3.27	Added support for the DLoG MPC5 CE device	2006-03-02
1.3.28	<p>When the machine is powered back on after automatic power-off it takes a while for the radio to reconnect and get it's IP address etc.</p> <p>Keyboard and scanner is now blocked while network connection is missing.</p> <p>Only works on Denso, Symbol and Intermec machines where the radio status can be checked.</p>	2006-03-16
1.3.29	The Exit Password dialog was not shown in some cases. This is an interim version only released for a specific project.	2006-04-20
1.3.30	<p>Added support for Unitech HT660.</p> <p>CEInstaller now correctly identifies a PSC Falcon 44xx that has Windows CE 5.0.</p>	2006-04-25
1.3.31	<p>Added support for Intermec CK31.</p> <p>Changed the licensing; now multiple network cards are supported.</p>	2006-05-04
1.3.32	<p>Added support for passthrough I/O redirection to/from serial port (DCS 1R ....ST) in ANSI emulation.</p> <p>Made some minor adjustments to Forced Mode Password.</p> <p>Denso BHT units had an unnecessary long retry time when waiting for attachment to an access point.</p>	2006-05-10
1.3.33	Major changes to Design Popup program and the popup	2006-05-23

Version	Changes	Date
	<p>keyboard itself to facilitate a new kind of keyboard layout supporting buttons with different up and down bitmaps.</p> <p>Added the Hide Mouse Cursor option so it is possible to get rid of the mouse cursor in touch-screen environments (regular Windows versions).</p> <p>When Access*One was started via Startup and the popup keyboard was configured to start hidden, Access*One lost focus which meant the user had to click on the screen before starting to type or scan.</p>	
1.3.34	<p>The 5250 function SaveScreen did not properly recreate the field for fields where it's display attribute had been overwritten by a non-attribute character after the field had been created.</p> <p>Changed the functionality of passthrough I/O redirection.</p>	2006-06-28
1.3.35	Fixed a bug in VT printing: if no data was sent between the sequence Controller Print On and Controller Print Off the printing progress statistic crashed with a divide by zero.	2006-06-30
1.3.36	Added support for Datalogic Kyman-net.	2006-07-07
1.3.37	Increased Automatic Power-off Time maximum value from 3600 to 2147483647 seconds. This should be more than enough.	2006-08-14
1.3.38	Added support for Psion Teklogix (tested on 7530)	2006-08-18
1.3.39	<p>Minor change to passthrough I/O redirection.</p> <p>If the Reader Port Name is empty/blank, the used comport is controlled by the numeric parameter nn in &lt;esc&gt;%nnR.</p>	2006-09-13
1.3.40	<p>Added the option "Disable Radio Integration".</p> <p>Changed serial printing/reader to not use DSR sensitivity on input as default setting.</p> <p>Added the option "Restore Screen in Line Mode"</p>	2006-10-05
1.3.41	Bugfix, Shift-d didn't produce the expected character D.	2006-10-19
1.3.42	Bugfix, in WinCE the popup keyboard lost it's topmost z-order when switching to the desktop.	2006-10-20
1.3.43	<p>Now stores the config version number in the config file</p> <p>Added diagnostics to display busy status for printing.</p> <p>Changed the logic for gathering input serial data when passthrough I/O redirection is active, we now wait a while (interval is baudrate dependant) for more data to arrive if we got at least one byte.</p>	2006-11-01
1.3.44	Removed all diagnostic and print progress info and changed	2006-11-06



Version	Changes	Date
internal	passthrough I/O to not wait for data, instead it send an empty response-frame every 100mS if no data is available.	
1.3.45 internal	Added a diagnostic messagebox if printing fails to create the spoolfile.	2006-11-08
1.3.46	'Productized' the changes to printing/passthrough I/O by adding config options to select wanted behaviour.	2006-11-10
1.3.47	Changes to CEInstaller to correctly identify Denso BHT 212. The 'C' key on BHT 200 now correctly maps to escape (0x1b).	2006-11-15
1.3.48	Added support for Symbol MC70	2006-11-24
1.3.49	Changes to passthrough I/O timing (custom project)	2006-12-01
1.3.50	Added support for Datalogic Skorpion and Symbol MC9090	2007-01-04
1.3.51	Fixed a bug related to the fix in 1.3.35 (the empty file was left opened resulting in a resource leak). After 509 empty prints, all file handles where exhausted and no more spoolfiles could be created.	2007-01-10
1.3.52	Added support for Symbol MC50. Added possibility to terminate a connection attempt by double-clicking on the screen.	2007-01-16
1.3.53	Bugfix for Datalogic Skorpion Autostart on warmboot didn't work. Installation directory moved to Backup	2007-01-22
1.3.54	Added externally configurable barcode string replacement so arbitrary barcode strings can be used as function keys.	2007-01-24
1.3.55	Added keyboard mode indicator for the FUNC key on Denso BHT-400.	2007-01-26
1.3.56	Removed the configuration of barcode identifier in Access*One for Symbol models. This allows external configuration in local application on the device or via Symbol Configuration Management.	2007-02-13
1.3.57	Added the possibility to disable/enable the scanner in ANSI mode via escape sequences.	2007-02-21
1.3.58	Added support for Datalogic Memor	2007-03-05
1.3.59	Added the option "Keep Alive Interval"	2007-03-09
1.3.60	Bugfixes: 1. Falcon 44xx with CE 5.0 was not correctly identified. 2. Not all machine configuration files (*.files) where correctly updated in 1.3.59.	2007-03-13
1.3.61	Datalogic changed radio module on Skorpion, so we had to remove the support for checking network AP attachment for Symbol Radio's.  Added support for "WLinqBell" in WIN32 version.	2007-03-22
1.3.62	Added support for Symbol WT4070/90.	2007-04-03

Version	Changes	Date
1.3.63	<p>Changed CEInstaller to use external file "ModelIdentify.txt" for CE 5 machines.</p> <p>Moved/added the StartMenuFolder= option from hardcoded to metafile [machine].files.</p> <p>Added option Common\Misc\Startup Delay Time.</p> <p>Fixed a problem with VT linemode when cursor was at right margin.</p>	2007-04-12
1.3.64	Added support for Intermec 700.	2007-04-13
1.3.65	<p>Added support for Denso BHT-282QW.</p> <p>Added support for advanced barcode editing via external module.</p>	2007-05-01
1.3.66	Added support for Freefloat Telnet Proxy Server.	2007-05-04
1.3.67	Support for RFC 4777.	2007-05-17
1.4.0	<p>Trimmed away old stuff (&lt; 1.3.x) from this version history.</p> <p>Better name for the installation image: accessone.exe</p> <p>Removed the option <b>Save MAC address in log</b> and the button <b>View MAC Log</b> from CEInstaller. This was done because MAC logging only worked for Denso devices and reported an error for all other devices. It still works for Denso devices.</p> <p>Replaced built-in popup keyboard with Freefloat Key*One, the version included is Key*One v1.0.10.</p> <p>Script keywords SetCanvas, ShowPopup, and HidePopup are now obsolete. Old scripts needs to be updated with the corresponding new script keywords for controlling the Key*One keyboard.</p> <p>Key mnemonic &lt;numclear&gt; replaced with &lt;clearfield&gt;.</p> <p>Some setting has changed place and name in the configuration file:</p> <p>The setting "Common\Screen\Popup\Show Popup Keyboard" is now called "Common\Keyboard\Show Keyboard".</p> <p>"Common\Screen\Popup\Snap Popup Keyboard" is now called "Common\Keyboard\Snap Keyboard".</p> <p>"Common\Screen\Popup\Popup Window Mode\Use Popup Window Mode" is now called "Common\Keyboard\Window Mode\Use Keyboard Window Mode".</p>	2007-06-26

Version	Changes	Date
	<p>"Common\Screen\Popup\Popup Window Mode\Font\*" settings are now located under "Common\Keyboard\Window Mode\Font\*".</p> <p>"Common\Screen\Popup\Popup Window Mode\Window\*" settings are now located under "Common\Keyboard\Window Mode\Window\*".</p>	
1.4.1	<p>Fixed small problem in integration with Key*One.</p> <p>In demo mode, the licensing dialog wasn't synchronous with Access*One if disconnected and reconnected.</p>	2007-06-27
1.4.2	<p>Added "Automatic Reconnect" option.</p> <p>Added the &lt;restart&gt; key definition which disconnects the current session and then immediately connects to a new session.</p> <p>Added the &lt;reboot&gt; key definition which reboots the PC. A new setting added for reboot type: Common/Win32/Forced Reboot (forced, non-forced).</p> <p>New setting: Common/Screen/Splash Screen Mode, can be set to Normal or Reboot.</p>	2007-07-10
1.4.3	<p>Added the option Display Characters in C1 Range for turning off parsing of control codes in the 0x80-0x9f range in ANSI/VT emulation. This enables the use, of code pages that has characters encoded in that range.</p>	2007-08-21
1.4.4	<p>The following keys and key combinations are now mapable:</p> <p>Left Shift Right Shift Left Control Right Control Left Alt Right Alt AltGr Apps Shift+Esc Pause Ctrl+Break</p> <p>Added support for the key Newline (5250).</p> <p>Added support for LXE HX2.</p>	2007-09-11

Version	Changes	Date
1.4.5	<p>Added support for Symbol MC9090G (CE 5).</p> <p>Added blocking of Windows key on selected CE-machines.</p> <p>Running Access*One unregistered in fullscreen mode on Windows CE, could result in the splash screen covering the registration dialog.</p>	2007-10-01
1.4.6	Updated the included Freefloat Key*One to version 1.0.11 which has key click support.	2007-10-04
1.4.7	<p>For some reason the mappings for some keys disappeared, affected keys were:</p> <p style="padding-left: 40px;">Apps Shift+Esc Pause Ctrl+Break</p> <p>In 3270/5250 only the escape key could be used for the reset function. Now if a key is mapped to "&lt;esc&gt;" it will also work as the reset function.</p>	2007-10-05
1.4.8	<p>Added support for Datalogic Pegaso</p> <p>Added indicator on status bar to show alpha/numeric mode on Symbol MC30xx.</p>	2007-10-15
1.4.9	<p>Problems with &lt;esc&gt; mapped to another key. KEYRESET added, it is defined to be the actual terminal reset function. KEYESC is now just a mappable physical key.</p> <p>KEYESC generates the terminal reset function, but now via a key mapping in 3270keymap.txt and 5250keymap.txt.</p>	2007-10-23
1.4.10	<p>Added settings for integration with Proxy*One.</p> <p>Updated the included version of Key*One to 1.0.12</p>	2007-11-05
1.4.11	<p>Win32 version now plays the same sound wavefiles (startup, normal, error &amp; barcode) as all other versions.</p> <p>Focus Time option added to address problems with other applications receiving focus on Windows startup.</p> <p>Moved settings for Proxy Server (address &amp; port) to the proxy client to allow for having multiple proxy servers. This enables systems to have one or more backup proxy servers.</p> <p>When a connect to the host via the proxy fails the user is asked</p>	2008-01-29

Version	Changes	Date
	<p>if a he/she wants to try to connect to the host directly, bypassing the proxy.</p> <p>The Startup Delay Time option is no longer a Windows CE only option. To activate the option, the command line parameter "-autostart" needs to be passed to Access*One.</p> <p>TN5250 Enhancement settings can now be specified in an external file.</p> <p>Bugfix: ALT-F4 can be used to exit Win32 version in fullscreen mode.</p> <p>Bugfix: Non-standard emulation screen size resulting in a smaller buffer than 80x24 caused a crash. Only affected VT/ANSI emulation.</p> <p>Changed the max number of loglines from 100 to 1000.</p> <p>Added support for Denso BHT-700 (BW/QW) models.</p> <p>Simplified determination of 1D and 2D scanner models.</p> <p>Modified order of radio initialization on Denso models to enable users to license Access*One without being attached to an access point.</p> <p>Fixed problem with BHT-400 shifted function keys. To access F5-F12 use the FUNC key to activate FUNC mode, then press F1-F4 or an arrow key to generate F5-F12.</p> <p>Added support for SCC SC600.</p> <p>Bugfix: [5250] SBA at row 1 Col 0 followed by a StartOfField is now handled.</p> <p>Bugfix: [5250] Defining a field that ended at the last buffer position resulted in a erroneous INVALID_ROW_COL_ADDR error code being sent to the host.</p>	
1.4.12	Updated the included version of Key*One to 1.0.13	2008-01-31
1.4.13	Removed the automatic machine detection from CEInstaller.	2008-02-13
1.4.14	Added support for Opticon PHL7000.	2008-02-19
	Added Intermec CN2B to supported devices.	

Version	Changes	Date
1.4.15	Bugfix Denso BHT400, Keys F1-F4 did not work correctly on the 50-key (full alpha) keyboard.	2008-03-25
1.4.16	On Windows CE, when running in full screen, the splash is now hidden when the Unregistered dialog is displayed.	2008-03-28
1.4.17	Bugfix Intermec, we no longer disable ScannerWedge if option Common/Barcode/Barcode Reader is disabled.  Added battery level indicator on the status line (CE devices).	2008-04-16
1.4.18	Added Intermec CN3A to supported devices.  Added battery level indicator on the status line also for regular WIN32.	2008-05-13
1.4.19	Fixed uni2ansi and uni2ebc to use full range 0x0000-0xffff	2008-08-28
1.4.20	Fixed problems with certain unicode characters that didn't work properly in ANSI linemode	2008-08-29
1.4.21	The config option "Raw Logging" also works on CE machines	2008-09-10
1.4.22	Updated the included version of Key*One to 1.0.22	2008-09-19
1.4.23	Yet another update of the included version of Key*One to 1.0.23 due to API interface change.	2008-09-29
1.4.24	Added support for Opticon PHL8000, Casio DTX7, and Motorola MC75.	2008-11-18
1.4.25	Added support for Mobilecompia M3 (MC-7500S)	2009-01-09
1.4.26	OEM version	2009-01-30
1.4.27	Added support for big text in VT/ANSI.	2009-03-11
1.4.28	FIX: changed field data validation in 5250 to use EBCDIC digits and characters	2009-03-24
1.4.29	FIX: Font Weight did not work.	2009-04-02
1.4.30	FIX: VT Answerback can be configured to contain any character using \xnn format	2009-04-20
1.4.31	FIX: Added support for key M3 on Denso BHT-700	2009-04-22
1.4.32	FIX: key F11 on MC70 (Windows Mobile) didn't work	2009-04-24
1.4.33	FIX: key F11 on Windows Mobile & Pocket PC didn't work	2009-05-19
1.4.34	Added option "Force DEC Special Graphics in G1"	2009-06-08
1.4.35	Added support for Mobilecompia M3 MC-7700S 2D	2009-07-22